

Tehisintellekt: sissejuhatatus





Boston Dynamics: Atlas 2016
<https://www.youtube.com/watch?v=rVlhMGQgDkY>



DARPA robotics challenge 2015
<https://www.youtube.com/watch?v=g0TaYhjpOfo>

Loengu teemad

- Orientatsioon ja taust
- Peamised meetodid
- Klassikalised näitesüsteemid ja valdkonnad
- Futuroloogia, hirmud, majandus

Orientatsioon

Marc Andreessen:

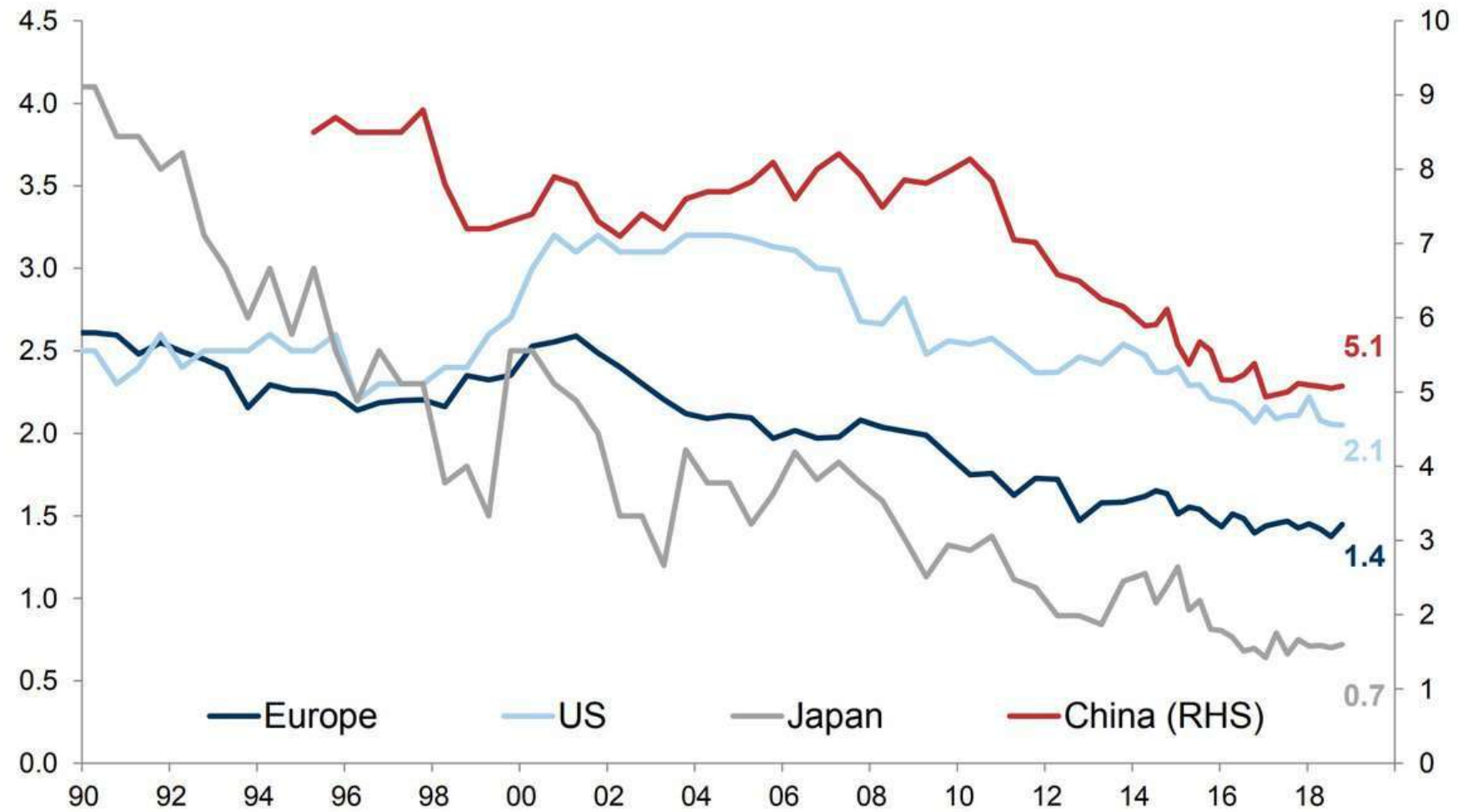
Software Is Eating the World

Paradoks:

IT ei ole sel sajandil globaalmajandust oluliselt efektiivistanud: vastupidi, GDP kasvutempo on pikaajalises langustrendis.

Exhibit 8: Long-term real global GDP growth forecast is at a historical low

Long-term (6-10y) GDP growth from Consensus Economics



Source: Consensus Economics, Goldman Sachs Global Investment Research

IT globaalne probleem ja lootus

Tööprotsesside automatiseerimine on väga raske: tarkvara ehitamine on aeglane ja kallis.

Tehismõistuse tehnoloogiad aitavad efektiivsemalt ehitada automatiseerimis-tarkvara.

Tugev ehk „päris“ AI, mida meil ei ole:

Masin on intellektilt inimloomusele lähedane või ületab seda, suudab täita tüüpilisi inimülesandeid, kasutada laia spektrit taustateadmisi ja omab mingil määral eneseteadvust.

Nõrk AI ehk see, mis tegelikult on:

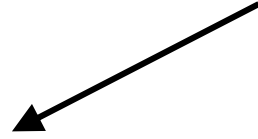
Tarkvara kasutamine spetsiifiliste probleemide või mõtlemisülesannete lahendamiseks, mis ei vaja laia spektrit inimvõimeid.

Yuval Harari:

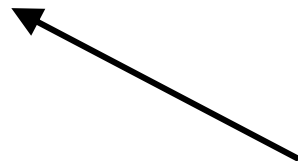
Automation will likely disrupt your current job (and your next one, and the one after that), and you'll be the target of attention-grabbing, behavior-modifying algorithms so exponentially effective you won't even realize you're being targeted.

Yuval Harari:

???



Automation will likely disrupt your current job (and your next one, and the one after that), and you'll be the target of attention-grabbing, behavior-modifying algorithms so exponentially effective you won't even realize you're being targeted.



Reaalsus

Paradoks:

Inimeste jaoks rasked ülesanded on masina jaoks tihtipeale lihtsad.

Inimeste jaoks lihtsad ülesanded on masina jaoks tihtipeale äärmiselt keerulised.

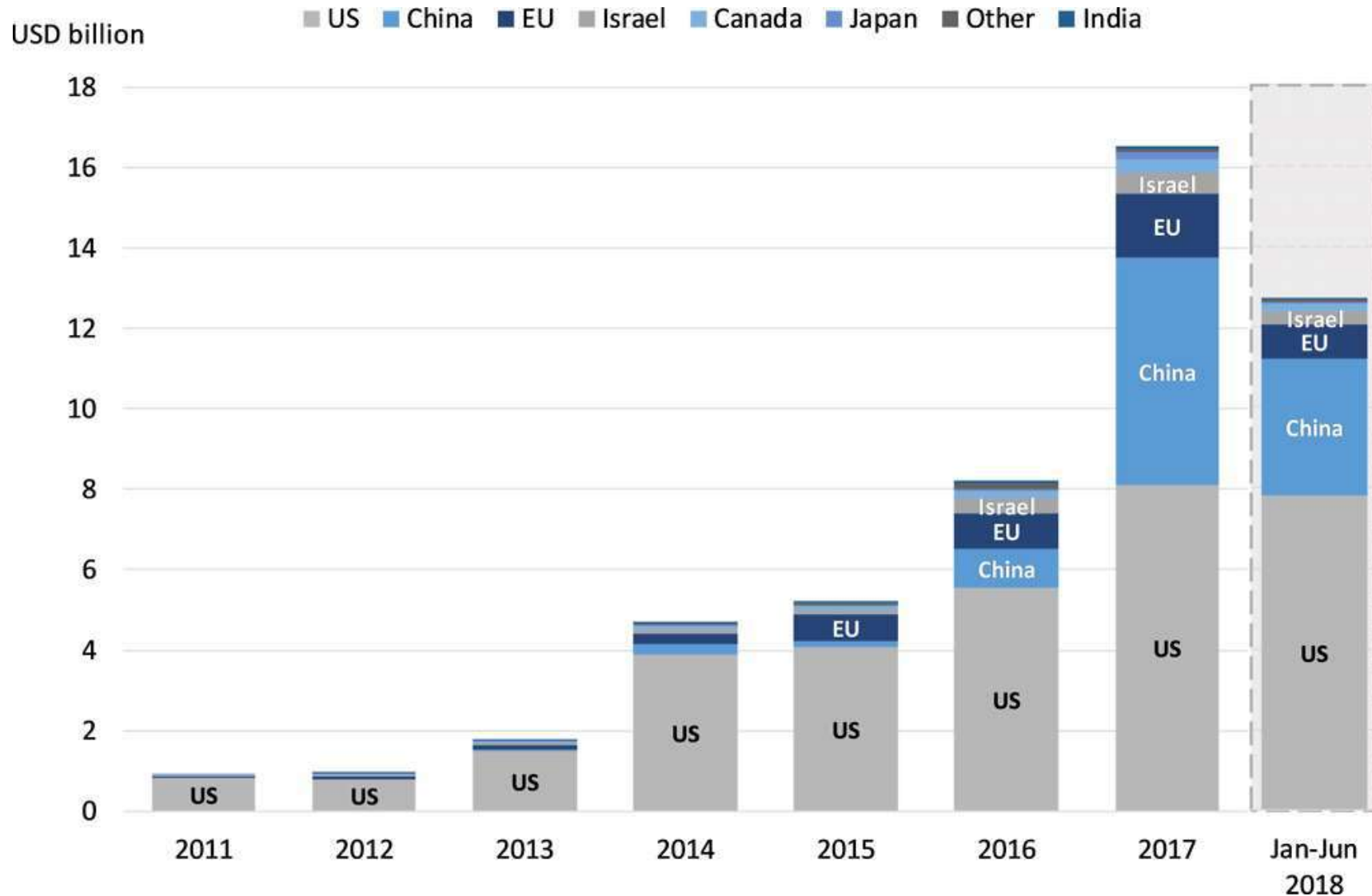
Inimene ei teadvusta oma ajutegevust.

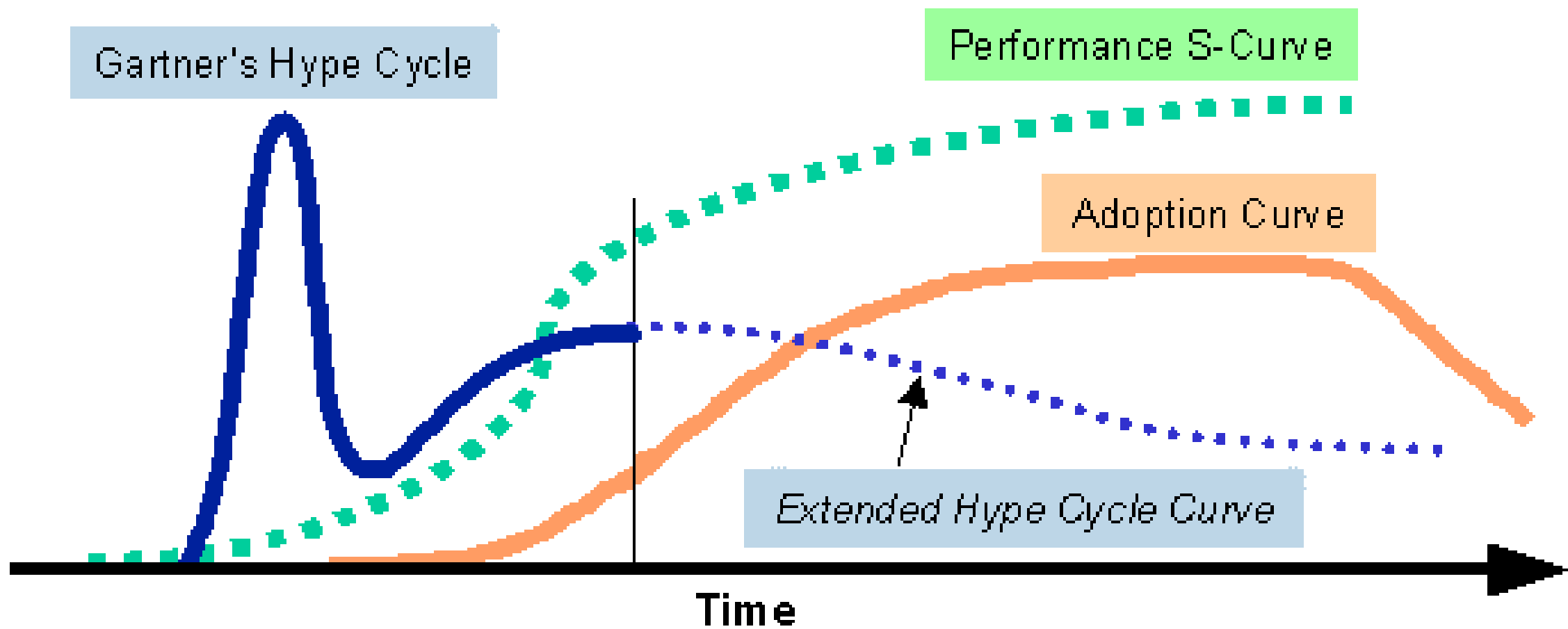
Inimesele paistab ekslikult, et paljud tema igapäevased toimingud on väga lihtsad ja ei nõua palju teadmisi ja ülikeerukaid ajuprotsesse

„Lihtne tehisintellekt“ ei ole võimalik.

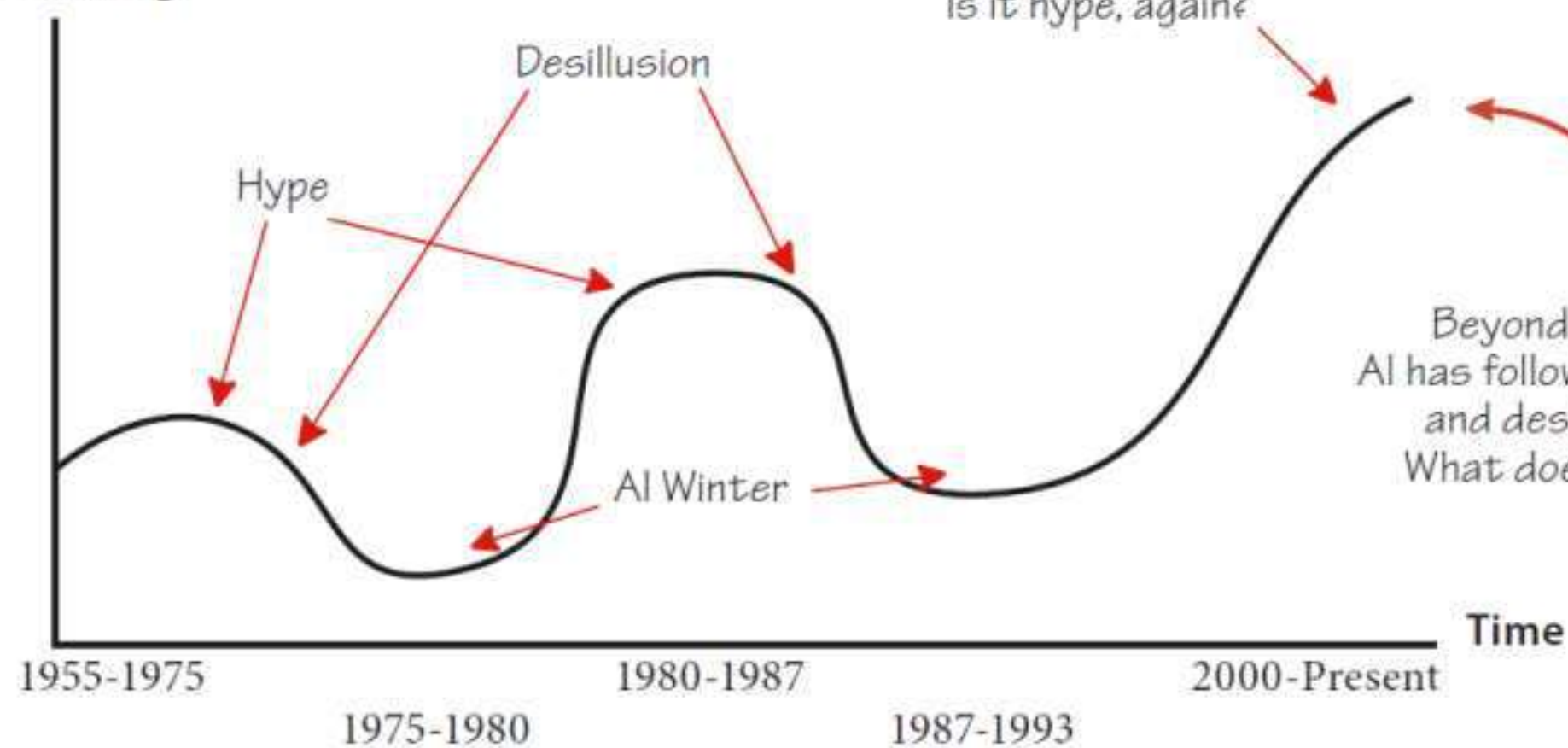
Süsteemi võimsamaks tegemiseks on paratamatult vaja lisada üha rohkem ja rohkem erinevate spetsiaalsustega komponente.

OECD: Total estimated investments in AI start-ups





AI funding



Beyond actual numbers,
AI has followed a pattern of hype
and desillusion for years.
What does the future hold?

Modern maxim:

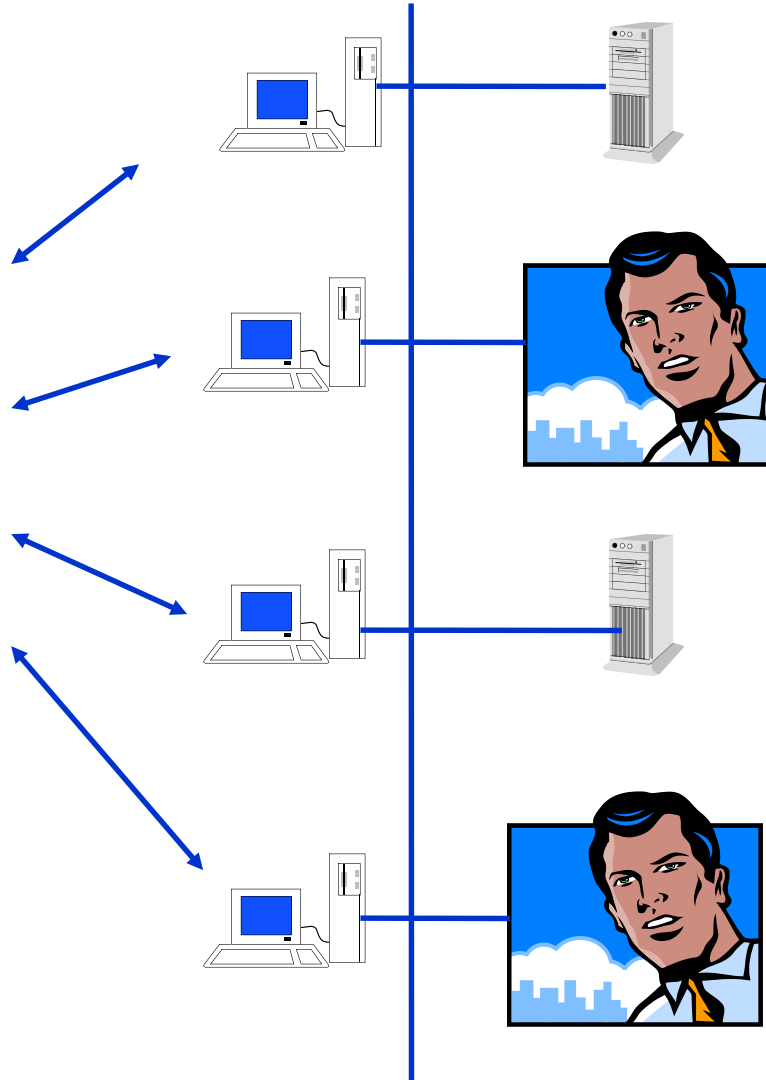
We tend to overestimate the effect of a technology in the short run and underestimate the effect in the long run

Turingi test

Turingi test: filosoofiline idee/ettepanek 50-datest



Mõistata, kas
chati-ekraani
taga on inimene
või programm?



Turing:

Kui katsetajad
ei suuda ära
arvata (st
ära-arvamise
sagedus on
50% ja 50%
eksitakse), siis
on jutlev
masin
päriselt
intelligentne.

Filosoofia ja praktika Turingi testi ümber

- Palju erinevaid vastuväiteid, et testi ärategemine ei tähenda veel “päris” intellekti.
- Ükski vastuväide ei ole ümberlükatav.
- Tegelikkuses ei suuda praegu keegi teha programmi, mis suudaks Turingi testi edukalt läbi teha: sellest ollakse väga kaugel.
- Praegu “edukalt” intellekti simuleerivad Turingi testi programmid on simulaatorid, mis hoiavad keskustelu väga kitsastes raamides ja sisuliselt ei saa üldse aru talle öeldud lausetest.
- Veidi põhjalikuma vestlusega saab alati aru, et simuleerival programmil pole üldist arusaama isegi igapäevasest elust ja asjadest ning ta ei oska probleeme lahendada.

Vestlev pseudointellekt: Eliza

- In 1966 **Joseph Weizenbaum** at M.I.T. wrote **ELIZA**, a famous program that simulates a Rogerian psychoanalyst by taking excerpts from the subject's comments and posing questions back to the subject.
- Eliza on lihtne programm, mille sees on suur hulk “standardlauseid” ja “standardsõnu”. Kui Elizale anda lause, milles on sõna tema “standardsõnade” loetelus, siis Eliza valib mõne “standardvastuse” ja paneb sinna sisse osi talle antud tekstist.
- Jääb mulje, et Eliza haakub vestlejaga. Eliza poolt antud laused on samuti harilikud inimkeelsed laused (sest need on programmeerija talle andmebaasi pannud).

Eliza: seletustega näitevestlus

ELIZA: HELLO, I'M ELIZABETH. WHAT WOULD YOU LIKE TO TALK ABOUT?

(the Welcome message)

USER: My mum is having a hard time.

ELIZA: TELL ME MORE ABOUT YOUR FAMILY

(line 5 changes 'mum' into 'mother', and this is matched at line 26; hence one of lines 28, 29, and 30 is randomly selected)

USER: There's just her, dad and my brother

ELIZA: WHAT DO YOU REMEMBER MOST ABOUT YOUR CHILDHOOD?

(line 6 changes 'dad' into 'father', and this is matched at line 27; hence either line 29 or 30 is randomly selected for display – line 28 is ruled out because it was the last response to be selected from this group)

USER: My father and brother were always quarrelling.

ELIZA: ARE YOU THE YOUNGEST IN YOUR FAMILY?

('father' is matched at line 27; hence either line 28 or 29 is randomly selected for display – line 30 is ruled out because it was the last response to be selected from this group)

USER: My brother is younger than me.

ELIZA: SO YOU ARE OLDER THAN YOUR BROTHER.

(line 31 matches, so line 32 then generates the response 'SO me IS OLDER THAN my brother'. Output substitutions then transform this in turn to 'SO YOU IS OLDER THAN my brother' (line 10), 'SO YOU IS OLDER THAN YOUR brother' (line 13), and finally 'SO YOU ARE OLDER THAN YOUR brother' (line 18).

Filosoofia ja praktika Turingi testi ümber

- Tegelikult Turingi testi edukalt tegev programm peaks:
 - Teadma igapäevaelust sama palju, kui keskmine inimene
 - Oskama lahendada ülesandeid
 - Saama aru vestluspartneri soovidest ja teadmistest jne
- See kõik on väga keeruline ja ei ole põhjust arvata, et lihtsakoeline programm sellega kuidagi hakkama saab.
- Kui programm sellega hakkama saab, siis on ta äärmiselt keeruline ja inimesed üldiselt ei saa enam aru, mis ta sees on ja kuidas ta töötab, ja pigem võiks arvata, et ta on siis ka päriselt intelligentne.

Filosoofia ja praktika Turingi testi ümber

Näiteid arusaamise raskustest:

The town councillors refused to give the angry demonstrators a permit because they feared violence. Who feared violence?

- a) The town councillors
- b) The angry demonstrators

The large ball crashed right through the table because it was made of Styrofoam. What was made of Styrofoam? (The alternative formulation replaces Styrofoam with steel.)

- a) The large ball
- b) The table

Mõned äärmus-lähenemised

Äärmused: putukad ja CYC

■ Putukad

- Rodney Brooks'i projekt MIT-s:
teha putuka analoogideks olevaid roboteid.
- Idee: kui oskame teha putuka intellektiga masinat, siis proovime sealt edasi liikuda väikese looma intellektiga masinani, sealt kõrgema loomani, sealt inimeseni ja edasi.
- Lootus: sel viisil ehitame mõistuse mehhanismid analoogiliselt nende tekkele looduses.



■ CYC

- Doug Lenati projekt CYC: teeme hiiglasliku andmebaasi faktidest ja reeglitest ja lihtsa järelduste tegemise programmi sinna peale.
- Idee: programm hakkab tekstidest aru saama, uusi reegleid ja fakte õppima, seejärel ka iseennast täiustama.
- Lootus: kvantiteet tekitab teatud piiri ületamisel kvalitatiivse hüppe.

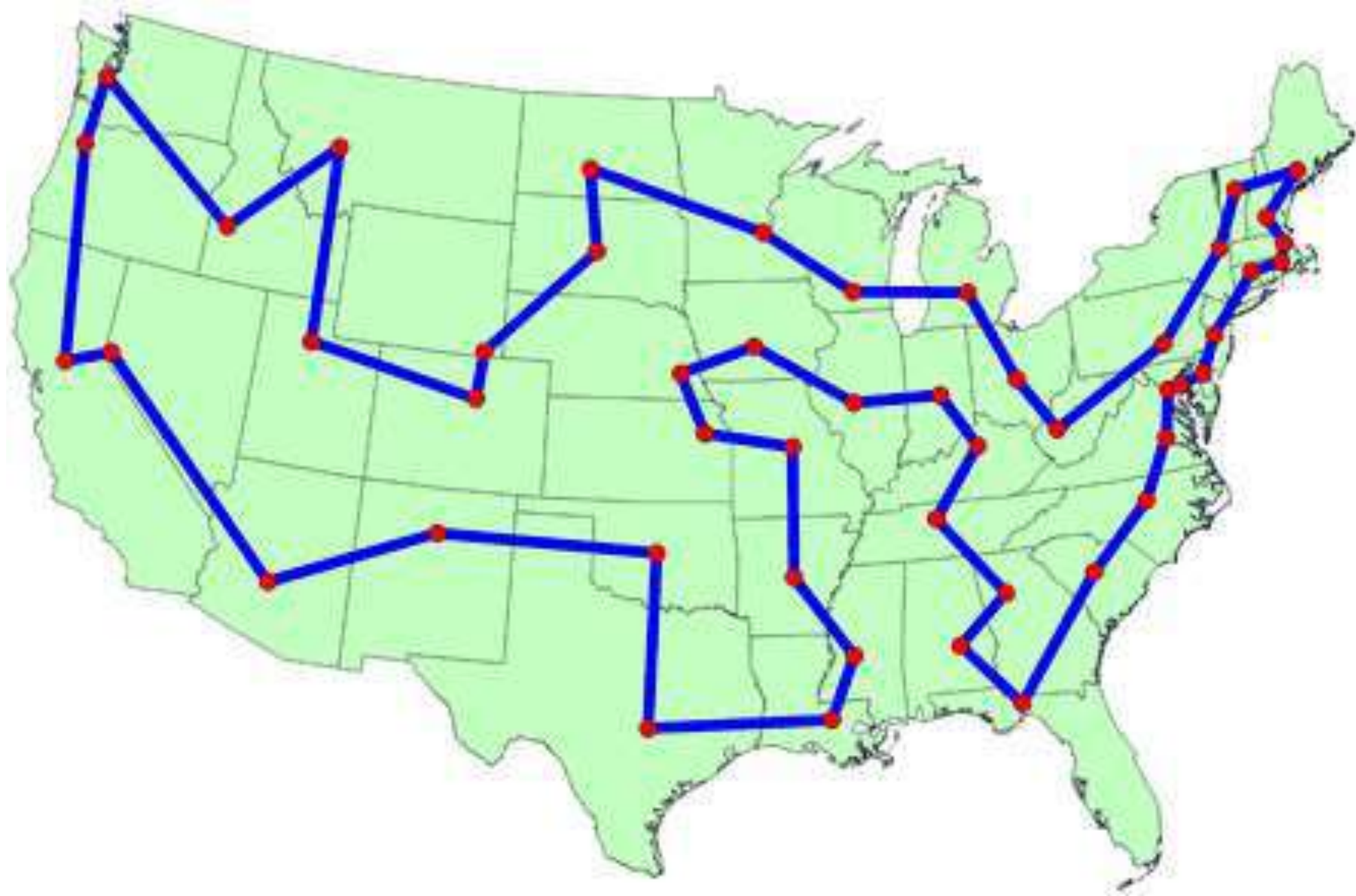
Peamised A.I. meetodid

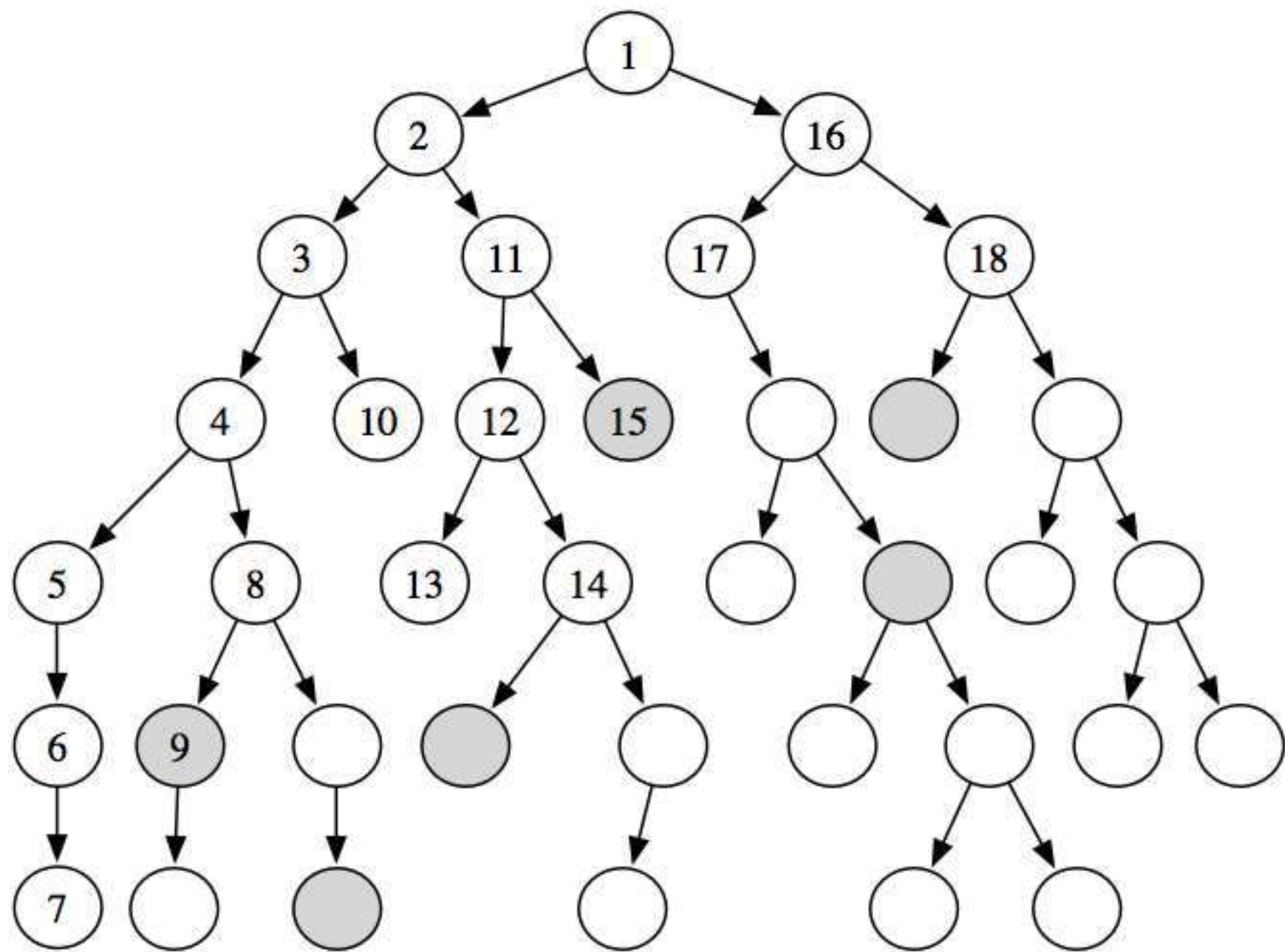
Kaks suurt AI meetodite valdkonda:

- **Otsing** paljude variantide seast a la malemäng.
- **Masinõpe** a la kasside ja koerte äratundmine pildilt, peale väga paljude õigesti sildistatud piltide näitamist.

Otsing

- Lihtsam variant: reeglid on alati samad
- Keerulisem variant: reeglid on iga ülesande jaoks erinevad

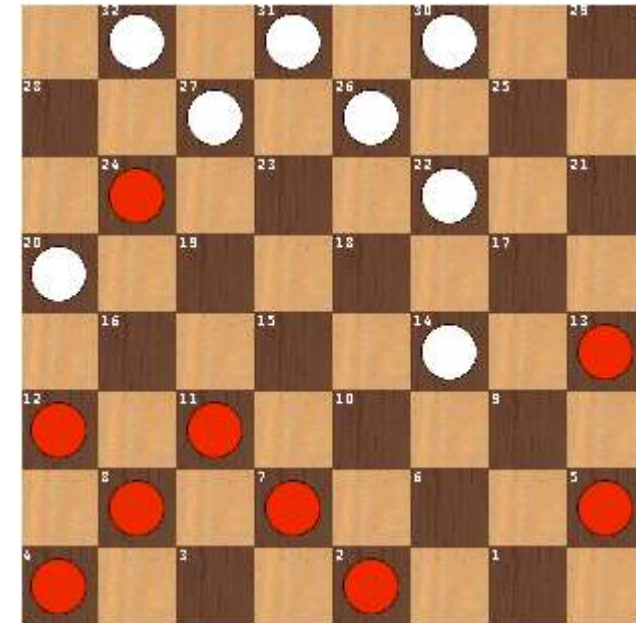




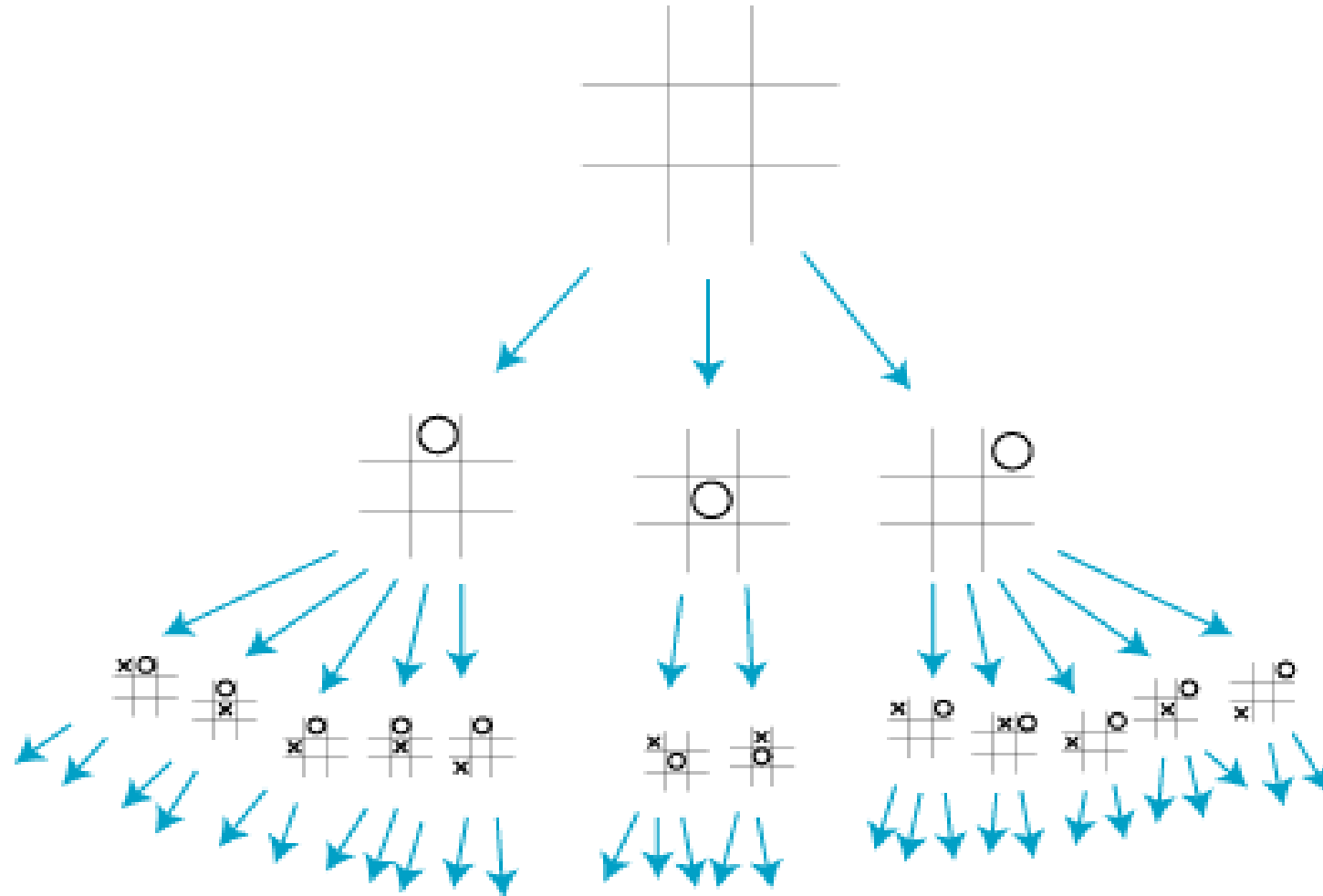
Inglise kabe lahendatud: 2007, Chinook UAlberta

Checkers has a search space of 5×10^{20}

Almost continuously since 1989, dozens of computers had been working around the clock to solve the game.



Mänguvariantide puu



Kuidas maleprogrammi teha?

- Teeme kõigepealt programmi, mis:
 - loeb seisu
 - teeb seisu järgi mällu tabeli kõigist võimalikest käikudest selles seisus.
- See ei olegi nii väga keeruline programm
 - Males keerulisem
 - Kabes lihtsam
 - Othello (reversi) veel lihtsam
 - Viis nuppu ritta: väga lihtne
- Nüüd võib programm tabelist juhusliku käigu valida.
- Ja juba mängibki! Kuigi kehvasti

Kuidas programmi veidi paremaks teha?

- Teeme eraldi väikese programmi, mis:
 - Võtab seisu ette
 - Ütleb umbes, kui hea seis on.
 - Seisu “headus” on number, mille meie programm arvutab.
- Kuidas öelda “kui hea”?
 - Vahel lihtne: kaotus / võit/ ei kumbki.
 - Numbrid: 1 -1 0
 - Keerulisem: “harilik maleseis”. Kumbalgi pole võitu ega kaotust.

Seisu hindamine

- Keerulisem olukord: harilik maleseis.
- Materjal:
 - Loeme kokku meie ja vastase nupud.
 - Lipp annab 10 punkti, vanker 6, jne.
- Positsioon:
 - Loeme kokku eelviimasel real etturid masinal/inimesel
 - Loeme kokku tsentriväljadel olevad etturid masinal/inimesel
 - Loeme kokku, mitmele tsentriväljale masin/inimene tuld annab
 - Vaatame üle kuninga kaitse tugevuse
 - jne
- Kokku saame seisu headuse numbri:
 - Masina materjal – inimese materjal + masina positsioon – inimese positsioon

Mis edasi?

- Nüüd on meil kõigi käikude tabel, iga käigu juures arvutame tekkiva seisu headuse numbri.
- Valime kõige parema tulemuse (maksimumi numbritest!) andva käigu.
- Programm hakkab veidi paremini mängima, aga:
 - Ikka mängib väga kehvasti!
 - Miks? Sest meie seisu headuse hindamise programm on nigel.
 - Kuidas headuse hindamist parandada?
- Loomulikult tuleks vaadata, mis käike vastane võib peale meie käike teha! Ja kuidas meie võime vastata. Jne.

Kuidas sellist seisude puud kasutada?

■ Eeldame lihtsalt, et:

- masin tahab teha käiku, mis annab kõige suurema headuse numbriga seisu.
- vastane tahab teha käiku, mis annab kõige väiksema headuse numbriga seisu.

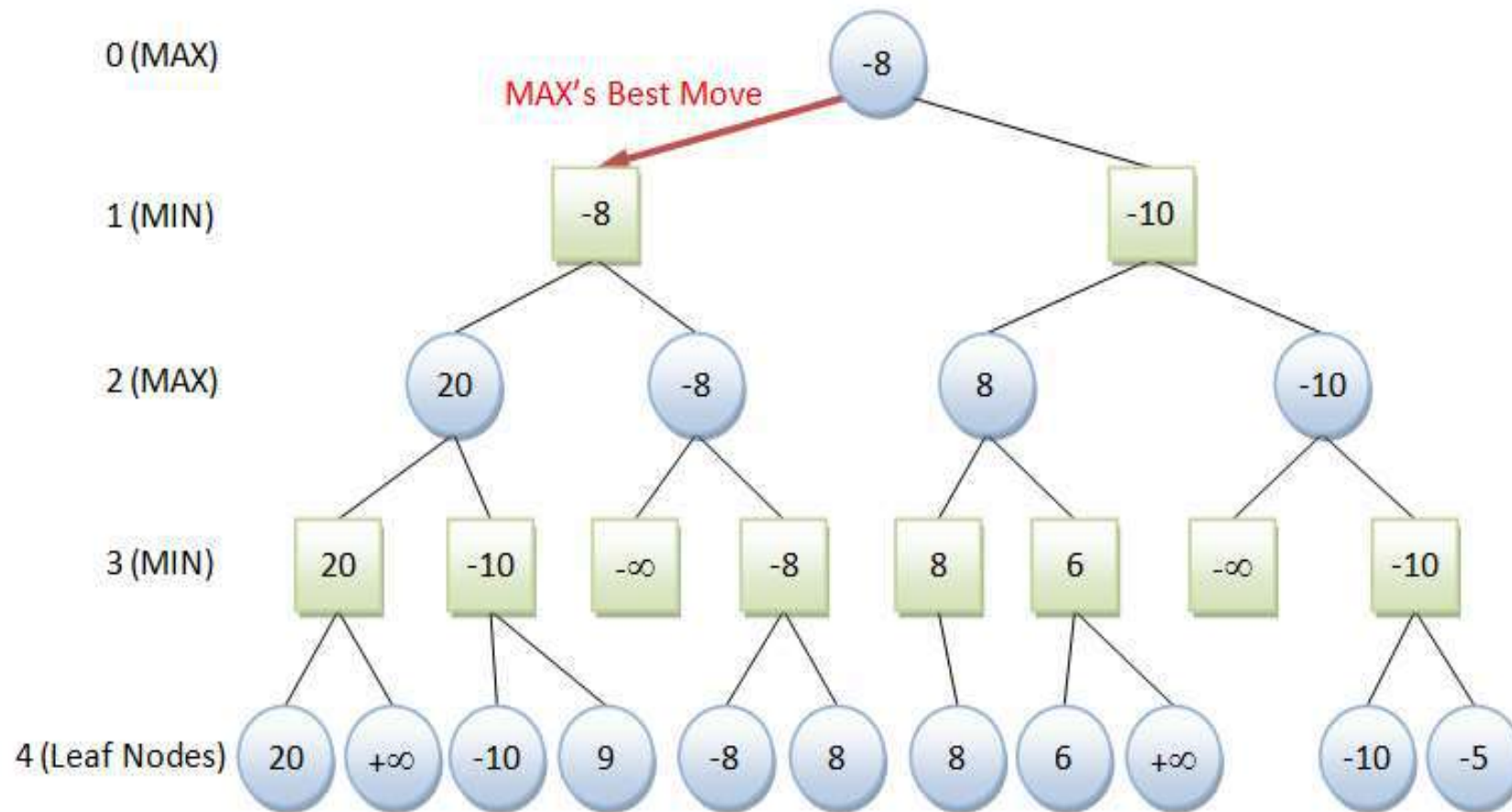
■ Seega:

- igast seisust valib masin käigu, mis on maksimum-headusega (masinale).
- igast seisust valib vastane käigu, mis on miinimum-headusega (masinale).

■ Idee:

- Vaatame käikude puud sügavuseni N (näiteks $N=3$)
- Kõige alumistel seisudel arvutame lihtsalt headuse välja
- Seejärel “tõstame” headuse numbreid ülespoole!

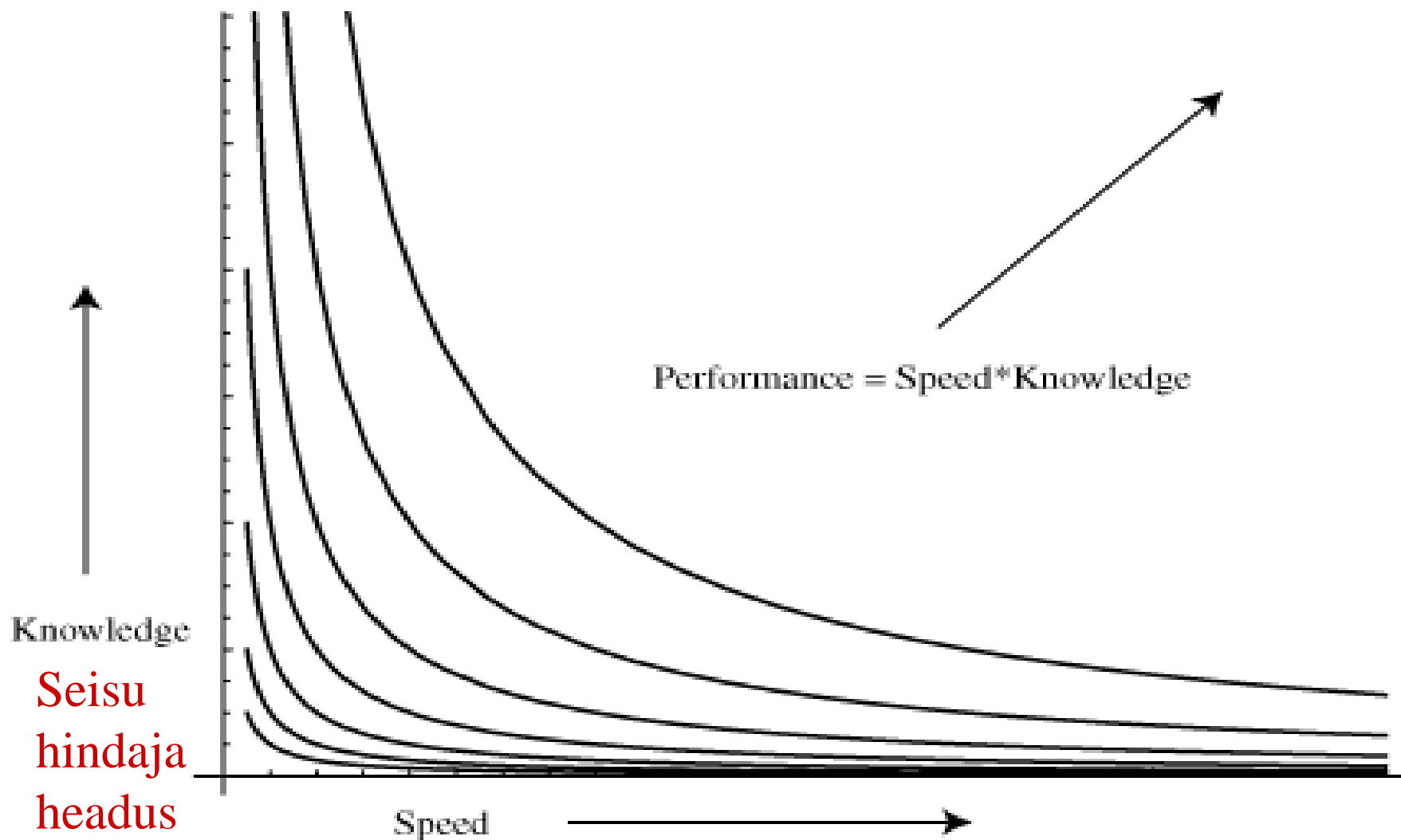
Otsing mängupuust: minimax algoritm



Kui heaks programmi saab?

- Mida sügavamad puud masin läbi jõuab vaadata, seda täpsemini ta käiku oskab valida.
- Puu läheb kiiresti väga suureks!
- Males ca 30 käiku ühes seisus.
 - **Esimesel tasemel käike 30.**
 - **Teisel tasemel käike $30 \cdot 30$**
 - **Kolmandal tasemel käike $30 \cdot 30 \cdot 30$**
 -
 - **N-ndal tasemel käike 30 astmes N.**
- Viiekümnendal tasemel oleks käike ca 30 astmes 50. See on rohkem, kui elementaariosakesi universumis!

Umbes nii: teadmised ja kiirus ja mängu headus



Seisu
hindaja
headus

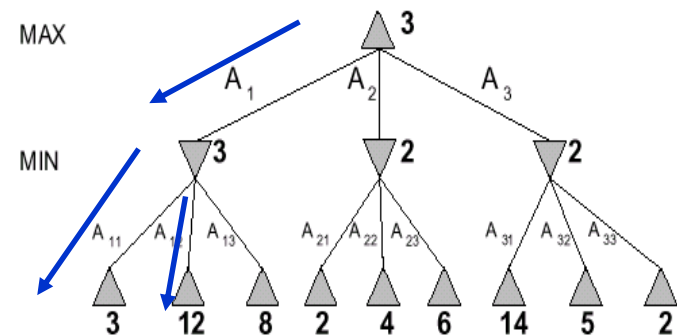
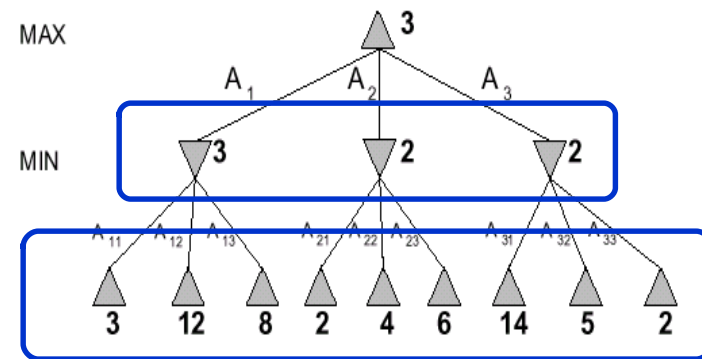
Kui suure puu jõuab läbi vaadata

Kuidas programmi parandada?

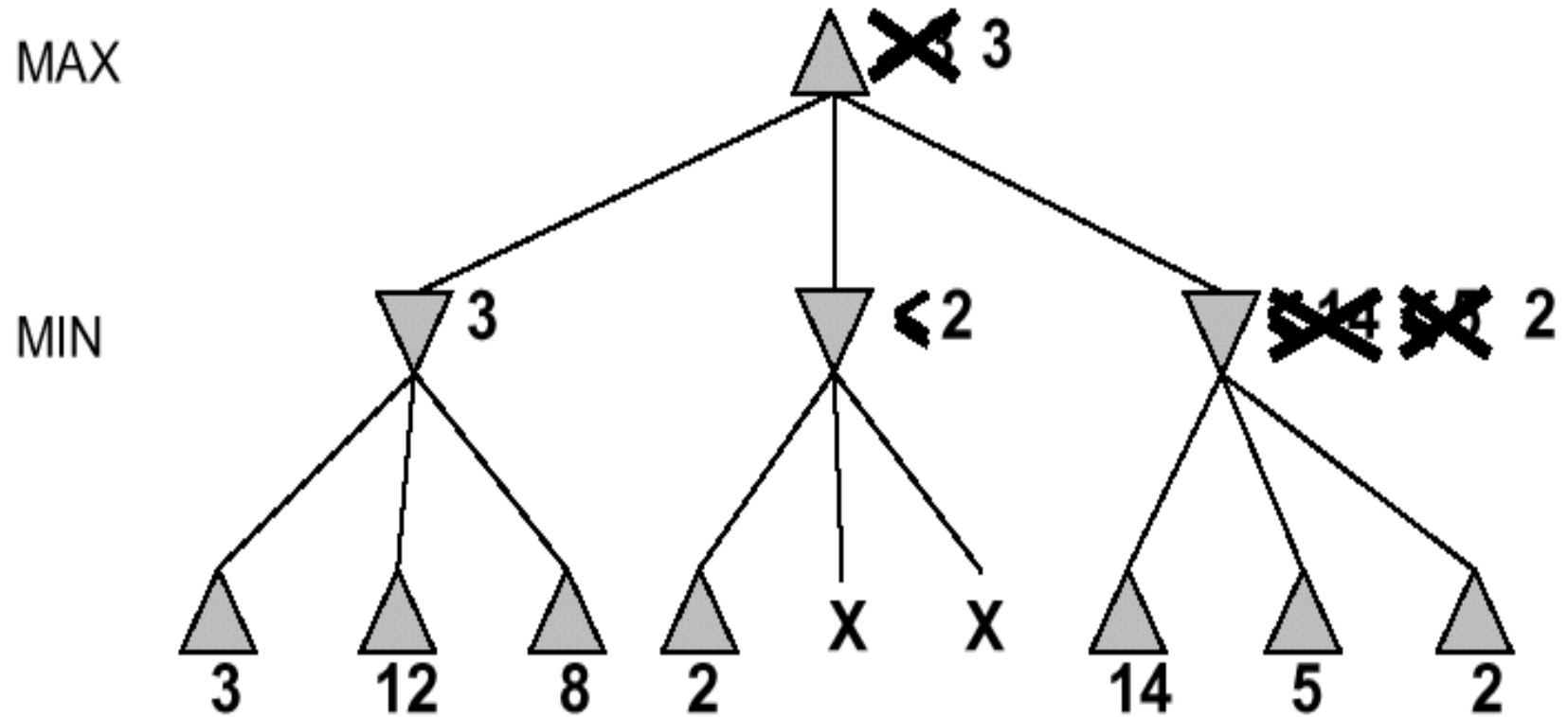
- Ei ole ühte head lahendust. On palju erinevaid nõkse!
- Näiteks:
 - Teeme seisu hindaja paremaks (programmi “targemaks”).
 - Aga siis läheb ta aeglasemaks ka.
 - Seega jõuame vähem käike läbi vaadata.
- Enamik nõkse on seotud käikude puu vähendamisega: ei ole vaja kogu puud läbi vaadata.

Laiuti vs sügavuti otsing

- Kaks võimalust:
 - Otsime puu läbi kiht-kihilt
 - Otsime puu läbi sügavuti, minnes alul vasakul maksimaalse sügavuseni
- Eelistatakse sügavuti otsingut!
 - Mälu vaja palju vähem
 - Muud eelised ka



Üks esimesi universaalseid meetodeid: alpha-beta



Baas-algorithm on lihtne otsing

```
int AlphaBeta (pos, depth, alpha, beta)
{
    if (depth == 0) return Evaluate(pos);
    best = -INFINITY;
    succ = Successors(pos);
    while (not Empty(succ) && best < beta)
    {
        pos = RemoveOne(succ);
        if (best > alpha) alpha = best;
        value = -AlphaBeta(pos, depth-1, -beta, -alpha);
        if (value > best) best = value;
    }
    return best;
}
```

Soteerimise nõks otsingu kiirendamiseks

- Sorteerida variandid seisust X enne otsingut ära: alustada tõenäoliselt paremate käikude proovimisega.
 - Suurendab tohutult alpha-beta efekti!
 - Kuidas sorteerida?
- Iteratiivne süvenemine. Teeme:
 - algul täisotsingu sügavuseni 2,
 - siis uue täisotsingu sügavuseni 4,
 - siis uue täisotsingu sügavuseni 6,
 - jne
 - iga kord kasutame eelmise otsingu tulemust sorteerimiseks!

Muud standardnõksud

- **“Killer moves”**: jätame otsides meelde eriti head käigud:
 - nii masinal kui vastasel
 - proovime kõigepealt varasemast meelde jäetud eriti häid käike
- **“Quiescence search”**: mõnda haru otsitakse sügavamalt:
 - ebastabiilses seisus otsime sügavamalt
 - stabiilses seisus otsime vähem sügavalt
 - otsime lõpuni kõik vahetused ja löögid
- **“Null-Move”**: mis siis, kui vastane saaks kaks käiku järjest?
 - Proovime nii, et vastane saab kaks käiku järjest
 - Kui on meile OK tulemus, siis see on positiivne faktor
 - Kui on meile halb tulemus, jätame meelde “killer move”

Kuidas optimeeringud mõjuvad?

■ Mida sügavam puu, seda suurem mõju. Näiteks:

■ Käikude puu sügavus viis:

- MiniMax: hindab 10,541,242 seisu
- Alpha-Beta: hindab 1,037,209 seisu
- A-B + “killer moves”: 530,587 seisu.

■ Käikude puu sügavus seitse:

- MiniMax: hindab ca 8,100,000,000 seisu
- Alpha-Beta: hindab 162,662,568 seisu
- A-B + “killer moves”: 46,455,262 seisu.

Täiendav idee: lõppmängude andmebaasid

■ Idee:

- ehitame hiiglasliku lõppmängu-seisude andmebaasi
- igal lõppmängu-seisul on andmebaasis öeldud täpne “headus” (võit, viik, kaotus)
- ehitamiseks: kõigepealt ühe nupuga lõppmängud, siis kahe nupuga lõppmängud, siis kolme nupuga lõppmängud jne.

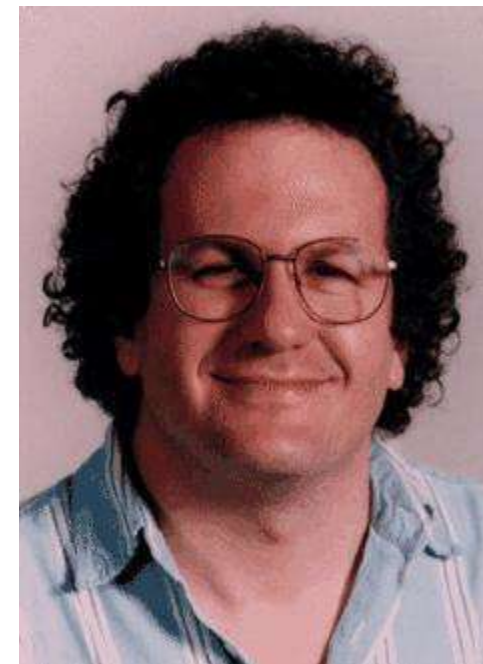
■ Kõigepealt tehti seda 8x8 inglise kabes:

- **Chinook**: Jonathan Schaeffer, Robert Blake, Paul Lu and Martin Bryant:
University of Alberta

- Kõigi seisude andmebaas, kus on 10 või vähem nuppu

■ Sügavad otsingud jõuavad tihti lõppmängude andmebaasini, kus on juba sees täpne seisuhinnang.

■ Otsing seega mõlemast suunast!



Praegune seis mõtlemismängudega

■ Lahendatud:

- Neli nuppu ritta, Qubic,
- Nine Man's Morris, Go-Moku,
- Awari
- Inglise kabe (8x8)

■ Väga tugevalt üle inimmängijate:

- Renju (viis nuppu ritta),
- Othello (Reversi), Scrabble, Backgammon

■ Selgelt üle maailmameistri taseme

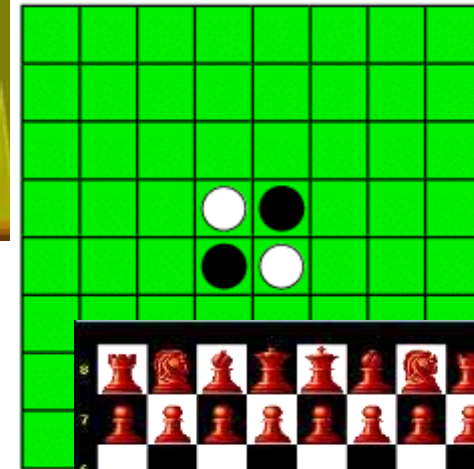
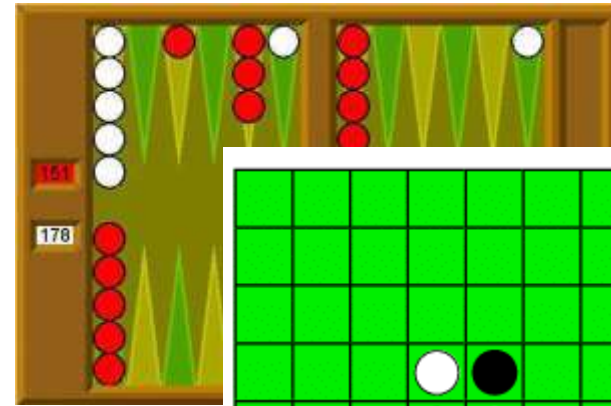
- Male
- Rahvusvaheline kabe (10x10)

■ Natuke üle maailmameistri taseme

- Go (19x19)

■ Tippmängija tasemel (???):

- Bridzh, Pokker



AlphaGo ja AlphaZero

DeepMind / Google projekt masinõppeks Go ja male jaoks.

Iseenda vastu mängides õpib seisumustrite järgi seisu headust kiirelt ja hästi hindama.

Hea seisuhindamine on kombineeritud Monte-Carlo otsinguga: vaatab juhusliku valikuga läbi hulga võimalikke käikude jadasid mängu lõpuni välja.

Olukord: iga ülesande jaoks omad reeglid

Hästi, kui õnnestub kirja panna väike hulk selgeid reegleid.

Halvasti, kui reegleid on palju:

- Ei jõua/oska neid kirja panna.
- Ei jõua otsida.

Näide: lahendamine loogikatõestuse abil

- Ülesanne tuleb loogika keeles formuleerida.
- Küsimus tuleb ka loogika keeles formuleerida.
- Tõestaja asub tõestust otsima.
- Kui anda lõpmatult palju aega ja mälu, siis tõestaja lõpuks ka tõestuse leiab (kui tõestus üldse teoreetiliselt eksisteerib). Keeruliste ülesannete puhul võtaks see lootusetult kaua aega (meenuta keerukusklasse!)
- Kui tõestust ei ole, siis tõestaja enamasti jääbki seda otsima, teadma, et sellist tõestust ei saa olla.

Vana matemaatikaprobleemi lahendamine A.I. abil

W. McCune 1996:

- The Robbins problem---are all Robbins algebras Boolean?- has been solved using his automated theorem prover EQP.
- Programm otsis lahendust ca üks nädal, kuni lõpuks leidis.
- Ülesanne oli matemaatikute poolt lahendamata, kuigi püstitati aastal 1933:

- Meil on antud Robbinsi algebra:

$$\begin{array}{ll}x + y = y + x. & \text{[commutativity]} \\(x + y) + z = x + (y + z). & \text{[associativity]} \\n(n(x + y) + n(x + n(y))) = x. & \text{[Robbins equation]}\end{array}$$

- Kas järgmised võrdused annava selle algebra jaoks vastuolu:

$$\begin{array}{l}x+y \neq x. \\n(x+y) \neq n(x).\end{array}$$

Tõestuse algus

2 [] $x+y=y+x$.

3 [] $(x+y)+z=x+y+z$.

4 [] $(x+y)+z=x+y+z$.

5 [] $n(n(n(x)+y)+n(x+y))=y$.

6 [] $n(x+y) \neq n(x)$.

64 [para_into,4.1.1.1,2.1.1,demod,3] $x+y+z=y+x+z$.

71 [para_into,5.1.1.1.2.1,2.1.1] $n(n(n(x)+y)+n(y+x))=y$.

73 [para_into,5.1.1.1,2.1.1] $n(n(x+y)+n(n(x)+y))=y$.

75 [para_into,6.1.1.1,2.1.1] $n(x+y) \neq n(y)$.

93 [para_into,71.1.1.1,2.1.1] $n(n(x+y)+n(n(y)+x))=x$.

116 [para_into,75.1.1.1,4.1.1] $n(x+y+z) \neq n(z)$.

130 [para_into,93.1.1.1.2,73.1.1] $n(n(n(n(x)+y)+x+y)+y)=n(n(x)+y)$.

132 [para_into,93.1.1.1.2,5.1.1] $n(n(n(x+y)+n(x)+y)+y)=n(x+y)$.

139 [para_into,116.1.1.1.2,64.1.1] $n(x+y+z+u) \neq n(y+u)$.

170 [para_from,130.1.1,73.1.1.1.2,demod,3,3] $n(n(n(n(x)+y)+x+y+y)+n(n(x)+y))=y$.

174 [para_from,130.1.1,5.1.1.1.1,demod,3,3] $n(n(n(x)+y)+n(n(n(x)+y)+x+y+y))=y$.

190 [para_into,132.1.1.1.1.1.2,64.1.1] $n(n(n(x+y+z)+y+n(x)+z)+y+z)=n(x+y+z)$.

211 [para_into,139.1.1.1,64.1.1] $n(x+y+z+u) \neq n(x+u)$.

253 [para_from,170.1.1,93.1.1.1.2] $n(n(n(n(x)+y)+n(n(x)+y)+x+y+y)+y)=n(n(x)+y)$.

260 [para_from,170.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z)+n(y+z))=z$.

346 [para_into,190.1.1.1.2,2.1.1] $n(n(n(x+y+z)+y+n(x)+z)+z+y)=n(x+y+z)$.

....

Tõestus jätkub

....

423 [para_from,253.1.1,5.1.1.1.1,demod,3,3,3,3] $n(n(n(x)+y)+n(n(n(x)+y)+n(n(x)+y)+x+y+y+y))=y$.

475 [para_from,260.1.1,73.1.1.1.2,demod,3,3] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+z+n(y+z))+z)=n(y+z)$.

486 [para_into,346.1.1.1.1.1.1.1,2.1.1,demod,3] $n(n(n(x+y+z)+x+n(z)+y)+y+x)=n(z+x+y)$.

684 [para_into,475.1.1.1.1.1.2.2,2.1.1] $n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z)=n(y+z)$.

801 [para_from,684.1.1,73.1.1.1.2.1.1,demod,3]

$n(n(n(n(n(n(x)+y)+x+y+y)+n(n(x)+y)+n(y+z)+z)+z+u)+n(n(y+z)+u))=u$.

848 [para_into,801.1.1.1.1,486.1.1,demod,3] $n(n(x+x+n(n(x)+x)+x)+n(n(x+x)+n(n(x)+x)))=n(n(x)+x)$.

878 [para_into,848.1.1.1.2,93.1.1] $n(n(x+x+n(n(x)+x)+x)+x)=n(n(x)+x)$.

897 [para_into,878.1.1.1.1.1.2,64.1.1] $n(n(x+n(n(x)+x)+x+x)+x)=n(n(x)+x)$.

948 [para_into,897.1.1.1.1.1,64.1.1] $n(n(n(n(x)+x)+x+x+x)+x)=n(n(x)+x)$.

1015 [para_from,948.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+x)+x+x+x)+x+y)+n(n(n(x)+x)+y))=y$.

1036 [para_from,948.1.1,73.1.1.1.2,demod,3,3,3] $n(n(n(n(x)+x)+x+x+x+x)+n(n(x)+x))=x$.

1053 [para_into,1015.1.1.1.2,423.1.1]

$n(n(n(n(n(x)+x)+x+x+x)+x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.

1079 [para_from,1036.1.1,93.1.1.1.2] $n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x)=n(n(x)+x)$.

1112 [para_into,1053.1.1.1.1.1,2.1.1,demod,3]

$n(n(x+n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.

1130 [para_from,1079.1.1,73.1.1.1.2.1.1,demod,3] $n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+y)+n(n(n(x)+x)+y))=y$.

1149 [para_into,1112.1.1.1.1.1,64.1.1]

$n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)$.

1169 [para_into,1130.1.1.1.2,174.1.1]

$n(n(n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)+x+n(n(n(x)+x)+x+x+x))+x)=n(n(n(x)+x)+x+x+x)$.

1211 [para_into,1169.1.1,1149.1.1] $n(n(n(x)+x)+n(n(x)+x)+x+x+x+x)=n(n(n(x)+x)+x+x+x)$.

1212 [binary,1211.1,211.1] \$F.

Mis on ekspertsüsteem

- Ekspertsüsteem on mõne spetsiifilise valdkonna jaoks kohandatud järelduste tegemise programm (näiteks variant teoreemitõestajast)
- Tihtpeale kasutavad ebaharilikku, spetsiaalselt valdkonnaga sobitatud loogikat.
- Enamasti sisaldavad paljusid valmiskujul reegleid.
- Enamasti sisaldavad mugavat kasutajaliidest: kasutaja ei pea loogika keelt oskama.
- Tüüpiliselt kasutatakse mõne suure rakenduse osana

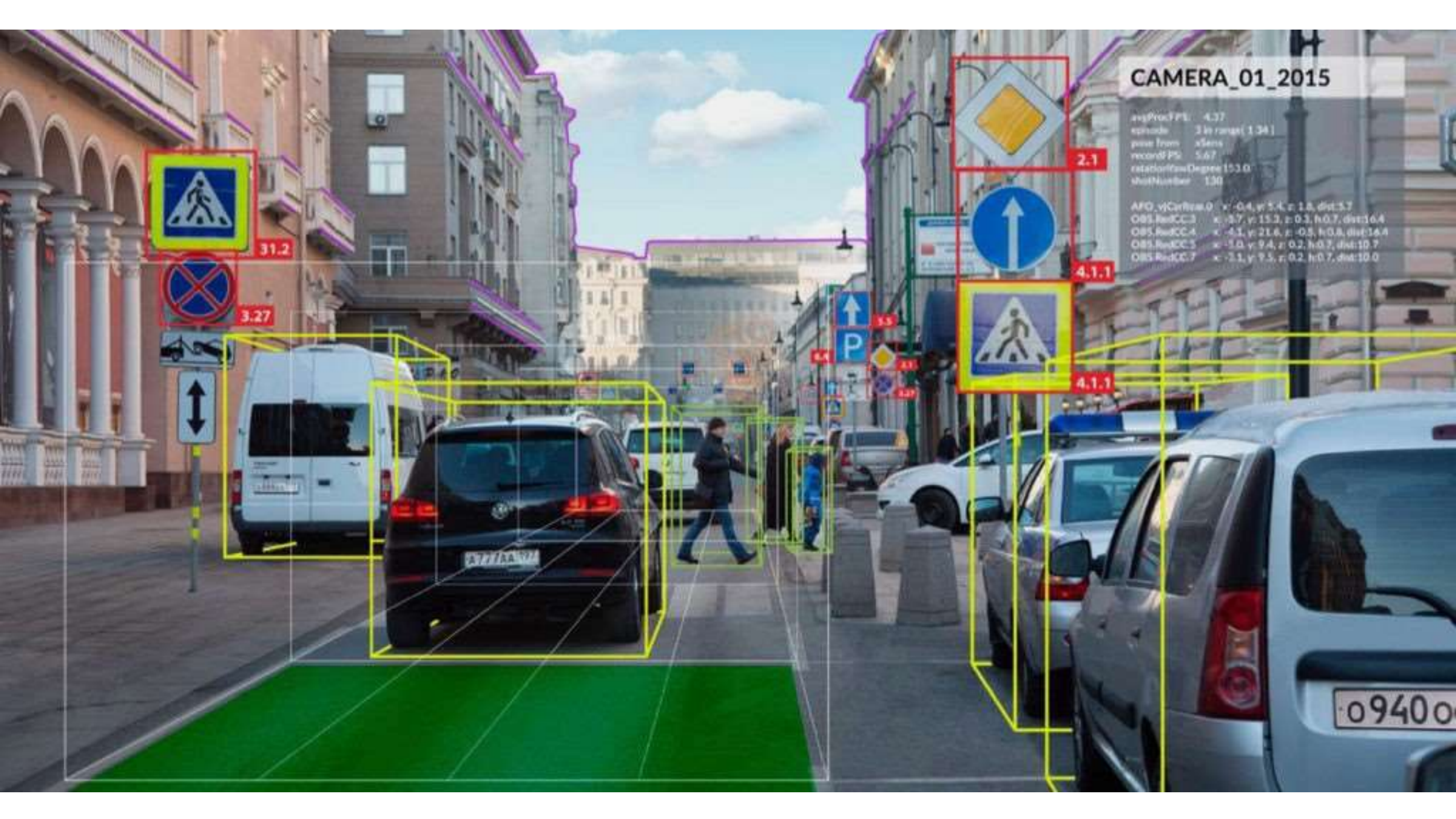
Loogika ja ebakindlad teadmised:
raske kokku panna

Suur hulk väga erinevaid tõenäosuslikke loogikaid.

Väga vähe reaalselt töötavaid süsteeme.

Õppimine

1. Klassikaline statistika
2. Gruppidesse jagamine (klasteriseerimine)
3. Sildistatud andmete pealt siltide ennustamine



CAMERA_01_2015

avgProcFPS: 4.37
episode: 3 in range(1, 34)
pose from: xSense
recordFPS: 5.67
rotationYawDegree: 153.0
shotNumber: 130

AFO_vjCarNew.0 x: -0.4, y: 5.4, z: 1.8, dist: 5.7
OBS_RedCC3 x: -0.7, y: 15.3, z: 0.3, h: 0.7, dist: 16.4
OBS_RedCC4 x: -4.1, y: 21.6, z: -0.5, h: 0.8, dist: 16.4
OBS_RedCC5 x: -5.0, y: 9.4, z: 0.2, h: 0.2, dist: 10.7
OBS_RedCC7 x: -2.1, y: 9.5, z: 0.2, h: 0.7, dist: 10.0



31.2



3.27



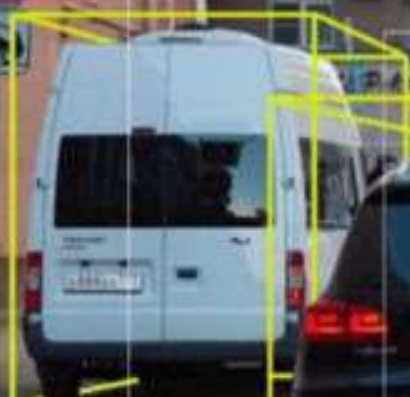
2.1

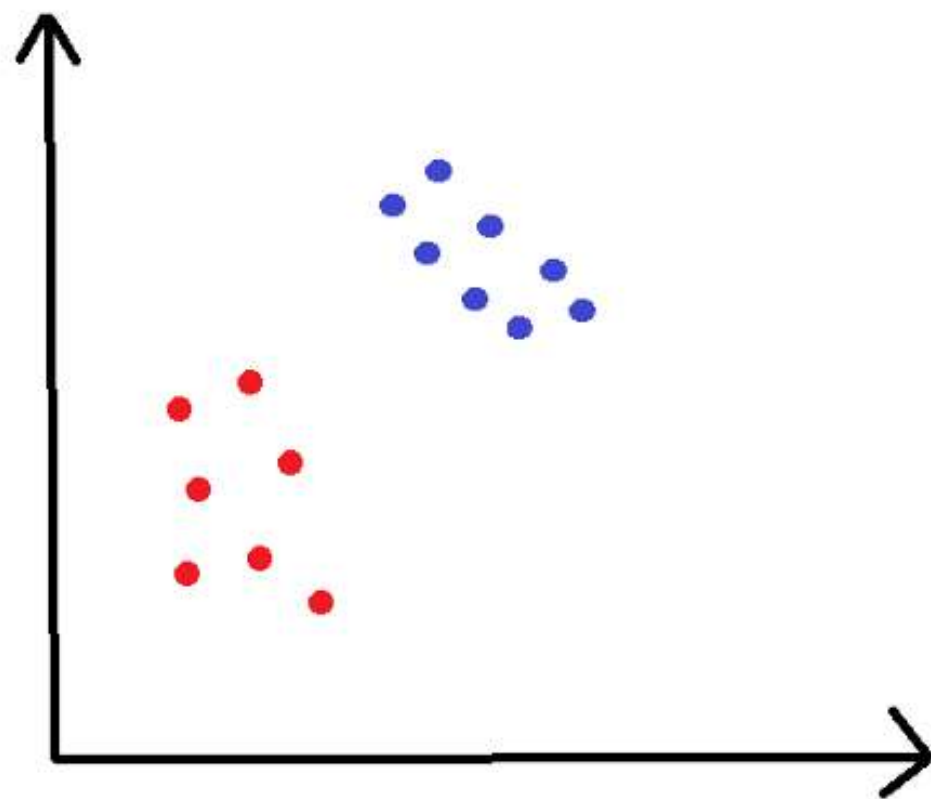


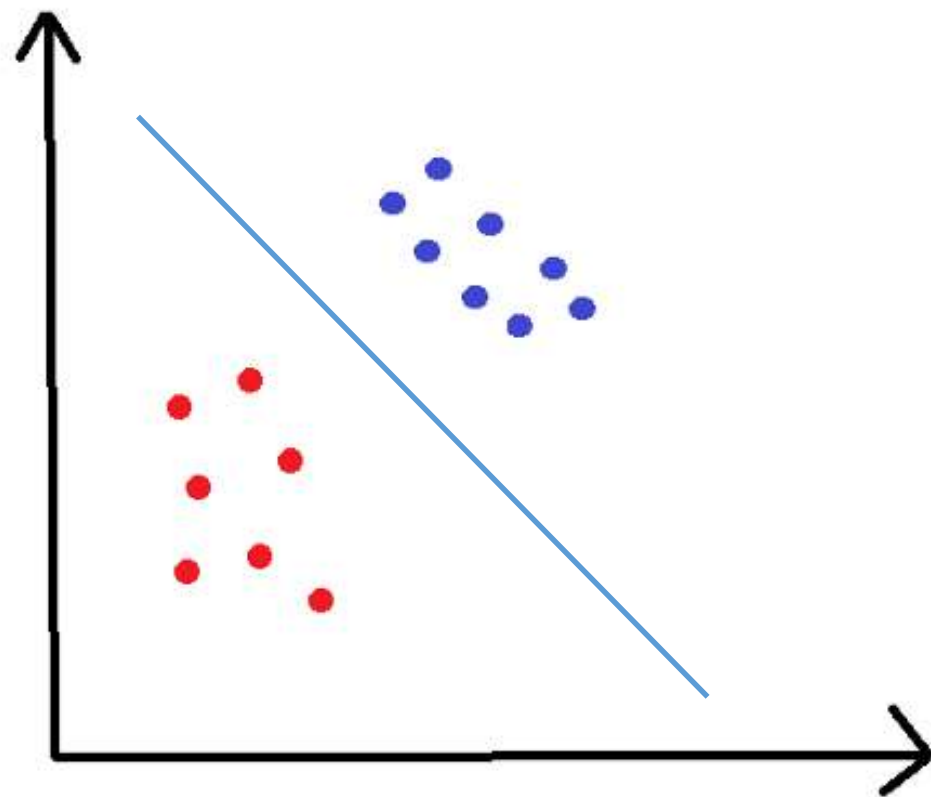
4.1.1

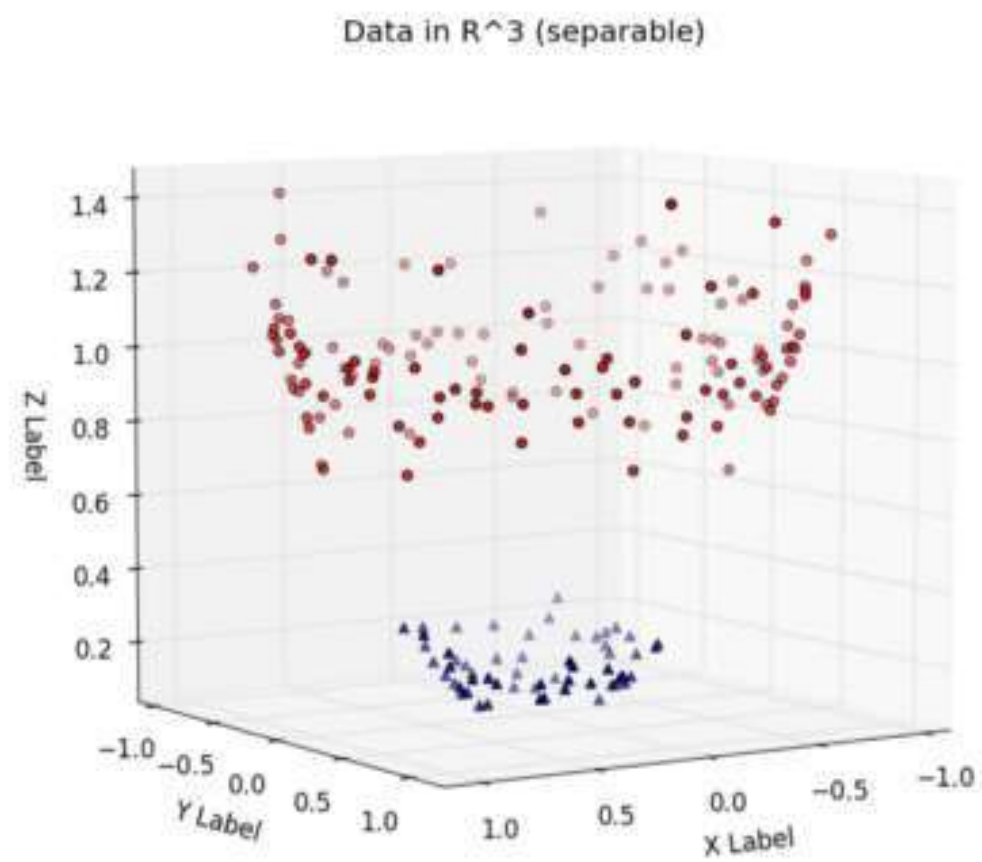
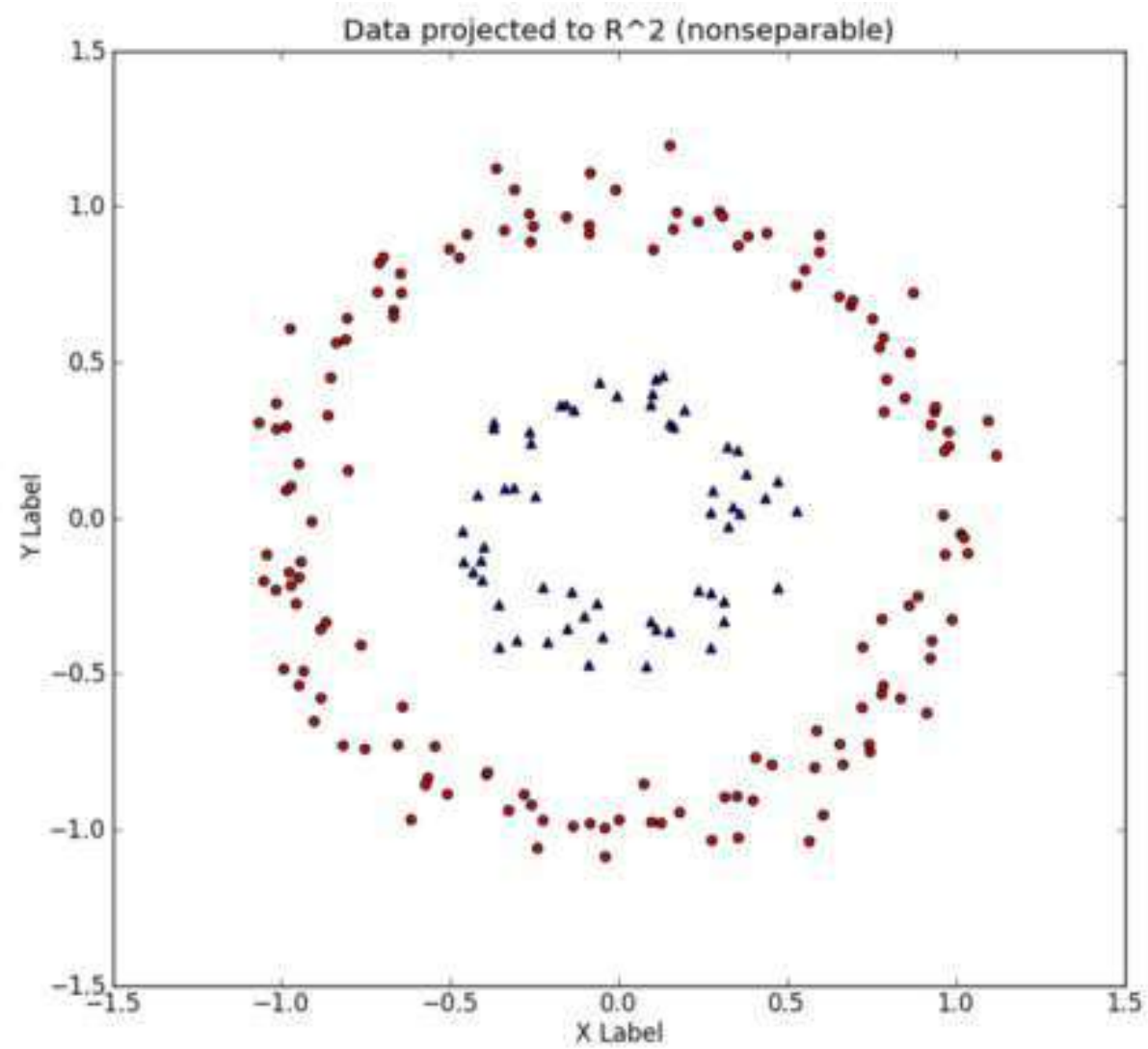


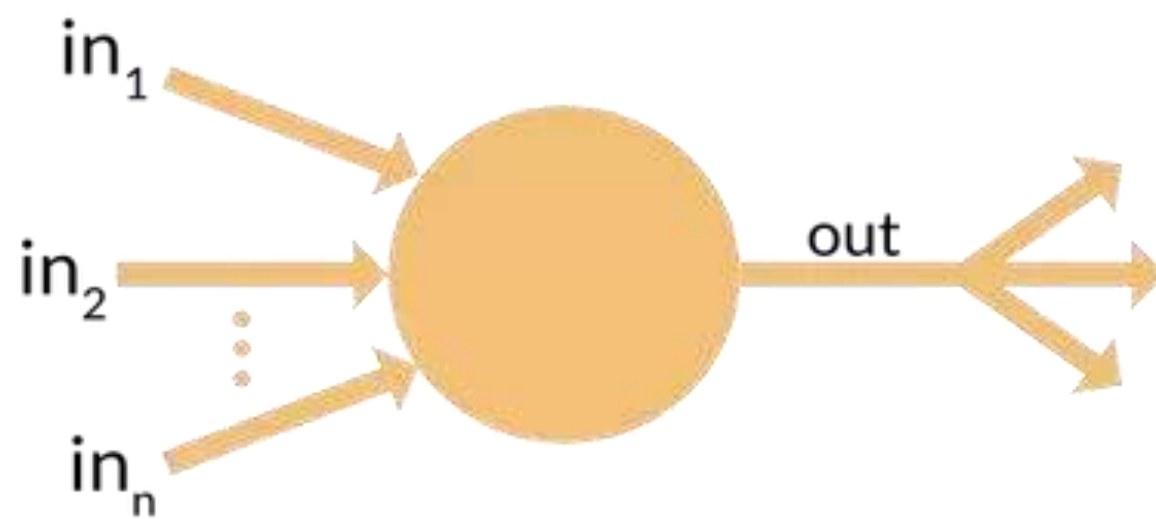
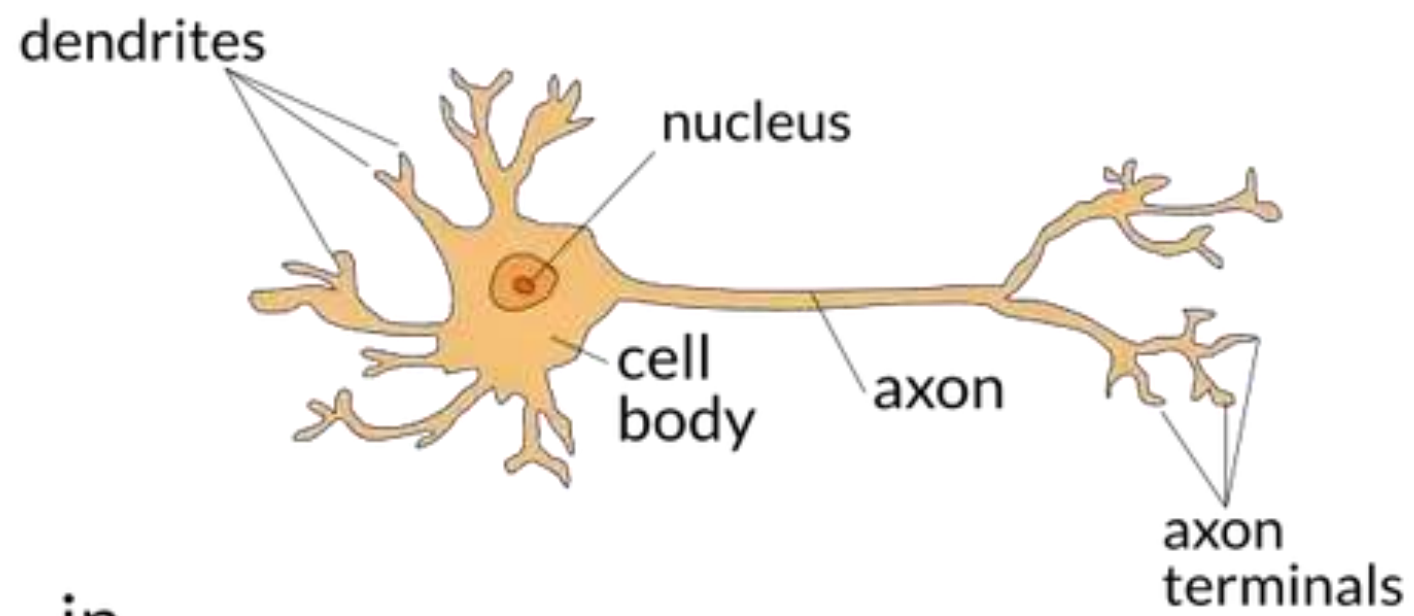
4.1.1

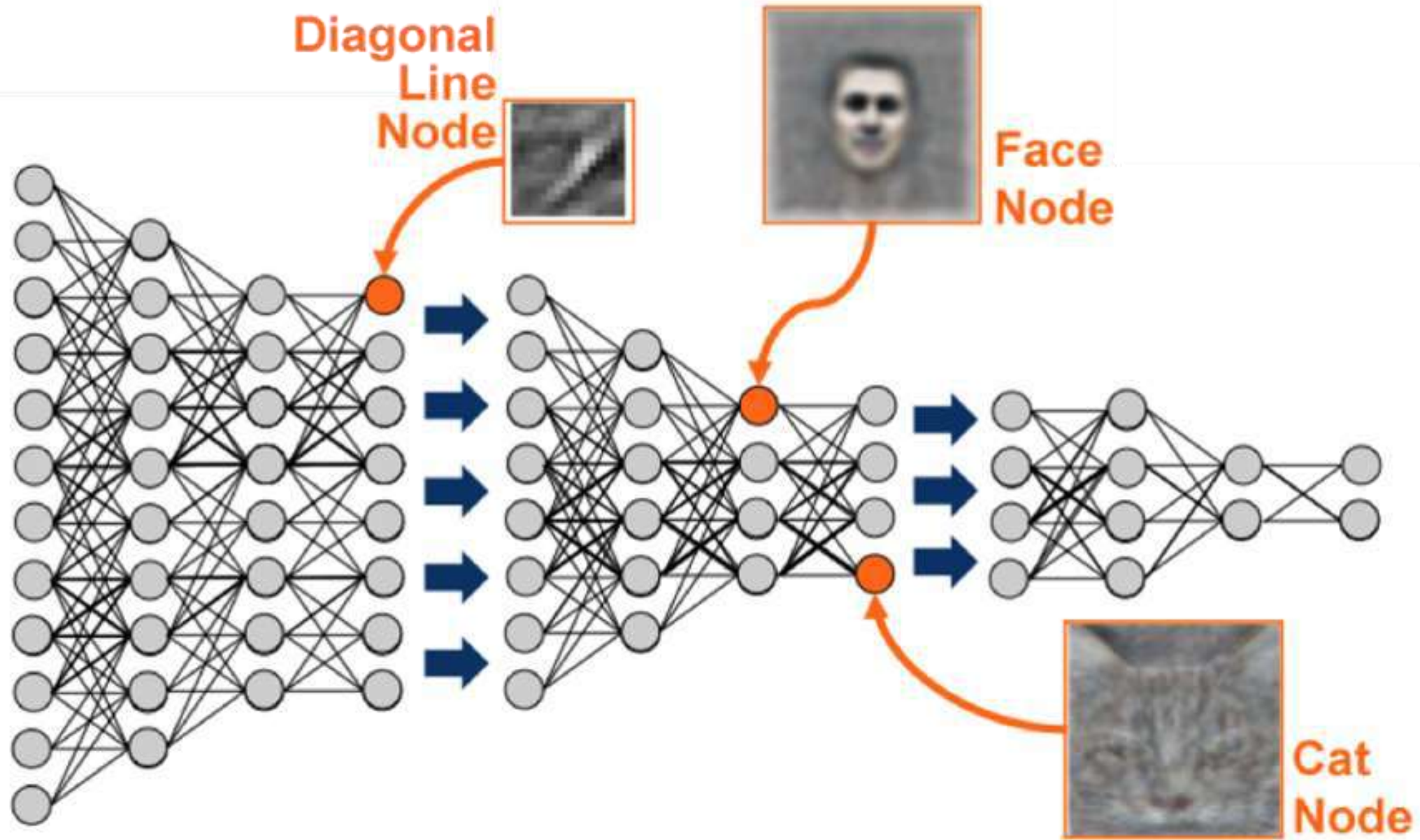




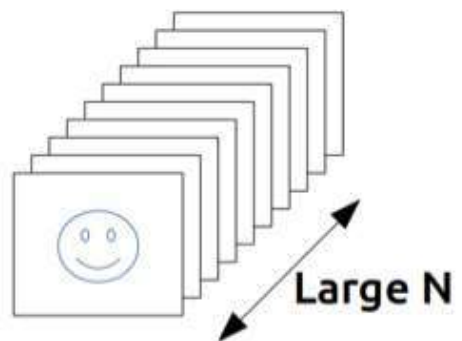




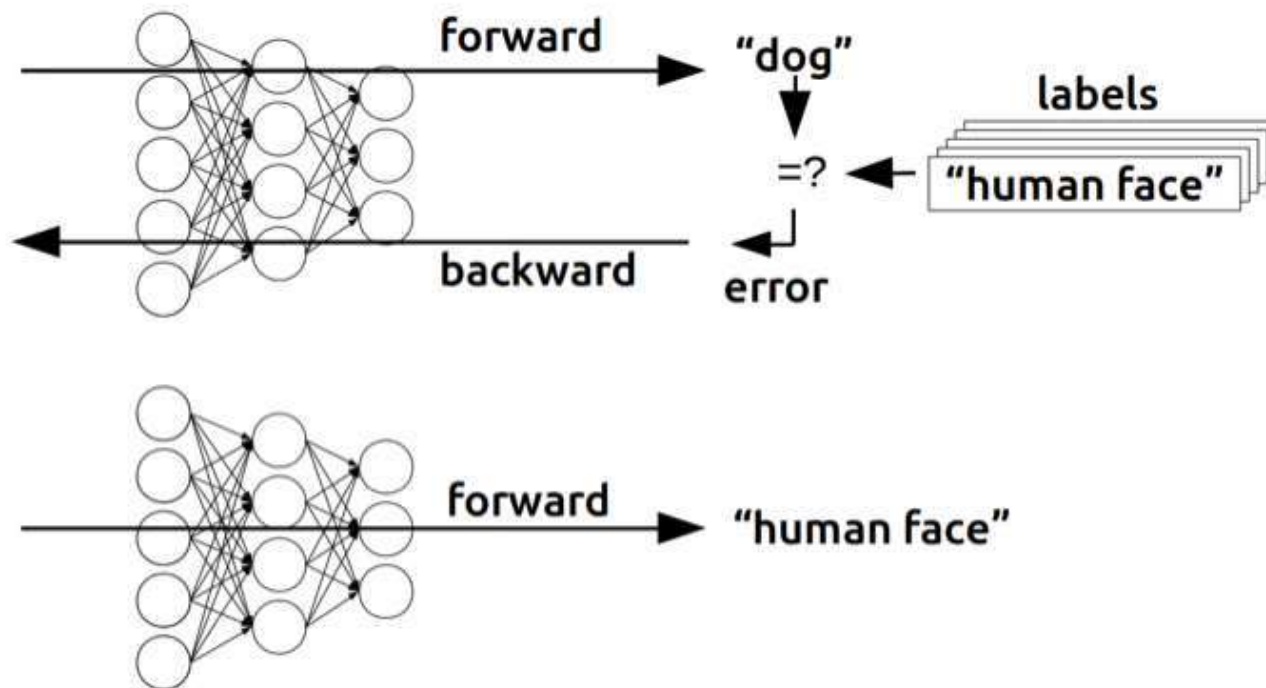
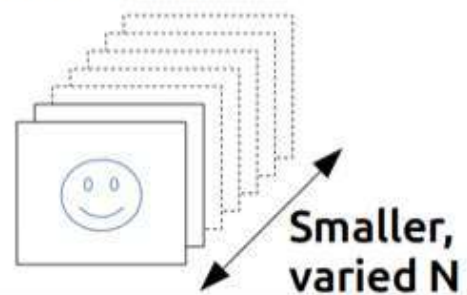




Training



Inference



Klassikalised süsteemid ja rakendused

IBM Watson võitis Jeopardy! aastal 2011




- It's just a bloody nose! You don't have this hereditary disorder once endemic to European royalty : *Haemophilia*
- You just need a nap! You don't have this sleep disorder that can make sufferers nod off while standing up : *Narcolepsy*
- Heitor Villa-Lobos dedicated his "12 Etudes" for this instrument to Andres Segovia : *Guitar*
- Paganini's "24 Capricci" set the standard for etudes for this instrument : *Violin*
- Rembrandt's Biblical Scene "Storm on the Sea of" this was stolen from a Boston museum in 1990 : *Galilee*

Masinõpe ja inimkeel: praegused tipp-süsteemid ei saa hakkama selliste ülesannetega:

- Linnavalitsus ei andnud meeleavaldajatele kogunemiseks luba, sest nad kartsid vägivalda. Kes kartis vägivalda?
- Palgil on kuus konna. Kaks lähevad ära ja kolm uut tulevad kohale. Mitu konna on nüüd palgil?
- Ma viisin eile oma riided pesumajja ja pean neile varsti järele minema. Kus mu riided praegu on?

Wolfram alpha online: 2009

 **WolframAlpha**[™] computational knowledge engine

meaning of life

Assuming "meaning of life" is a quantity | Use as referring to English words or a movie instead

Input interpretation:
answer to life, the universe, and everything

Result:
42
(according to Douglas Adams' Hitchhiker's Guide to the Galaxy)

Computed by: [Wolfram|Mathematica](#) | Download as: [PDF](#) | [Live Mathematica](#)

Masinõpe ja NLP

Teksti jätkumist ennustavad süsteemid:

Bert (Google)

GPT-2, GPT-3 (OpenAI)

Robotika

Süntees kõigist A.I. valdkondadest, pluss
mehaanika

Iseõitvad autod

2005

Stanfordi Stanley
võidab DARPA
Grand Challenge II



Darpa urban challenge 2007

96 km linnas, 4:10

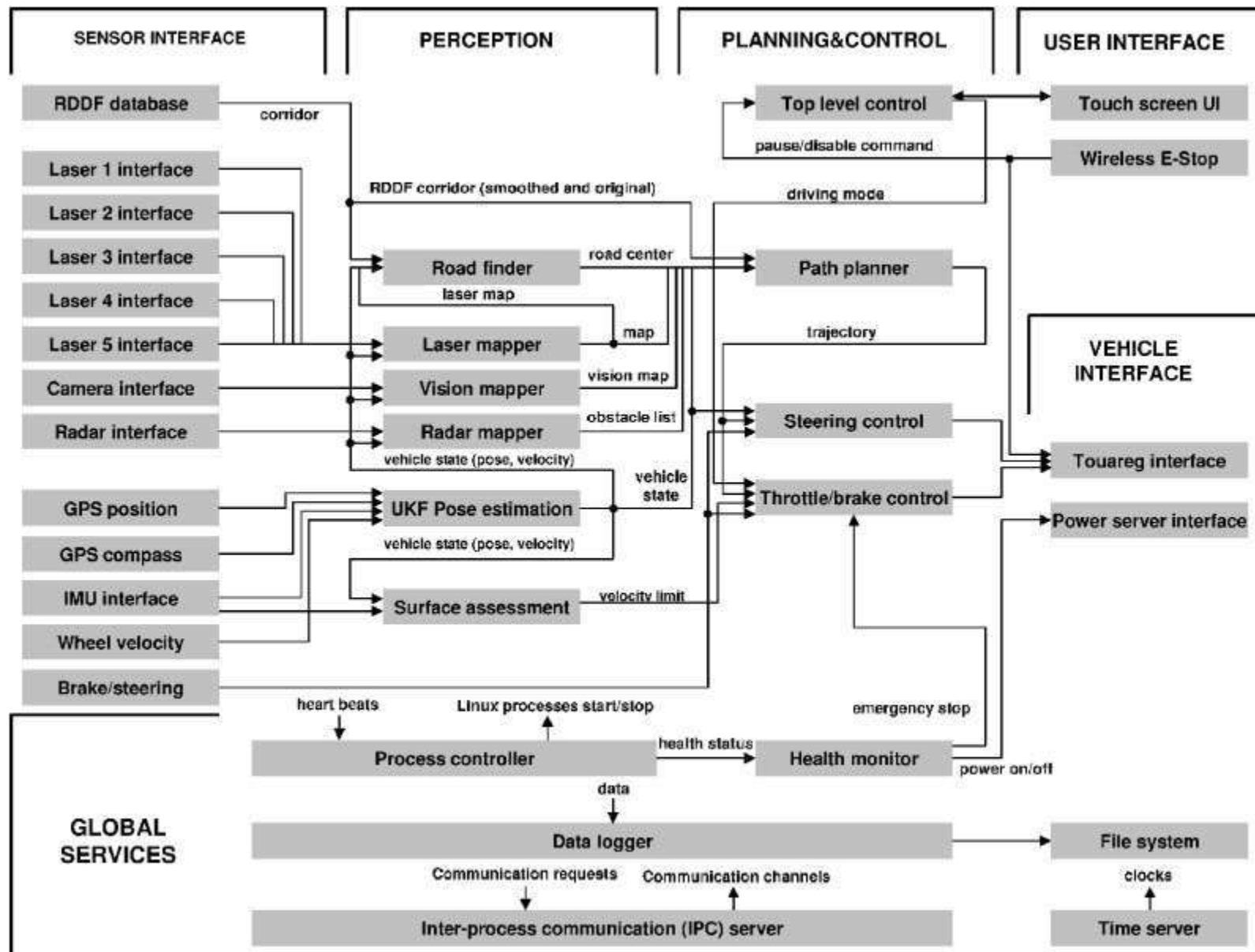


Google autonomous car

Based on vehicle Stanley which won the 2005 DARPA Grand Challenge.

In August 2012, the team announced that they have completed over 300,000 autonomous-driving miles (500 000 km) accident-free





Iseõitvad autod on keerulisemad, kui algul paistis

Chris Urmson, the former leader of Google's self-driving car project, once hoped that his son wouldn't need a driver's license because driverless cars would be so plentiful by 2020.

Now the CEO of the self-driving startup Aurora, Urmson says that driverless cars

„will be slowly integrated onto our roads over the next 30 to 50 years“.

Tesla

„Traffic-Aware Cruise Control **cannot detect all objects and may not brake/decelerate for stationary vehicles**, especially in situations when you are driving over 50 mph (80 km/h) and a vehicle you are following moves out of your driving path and a stationary vehicle or object is in front of you instead.”

Volvo

„In other words, **your Volvo won't brake to avoid hitting a stopped car that suddenly appears up ahead**“.

iRobot
Roomba

algus 2002

käivitaja
Rodney Brooks

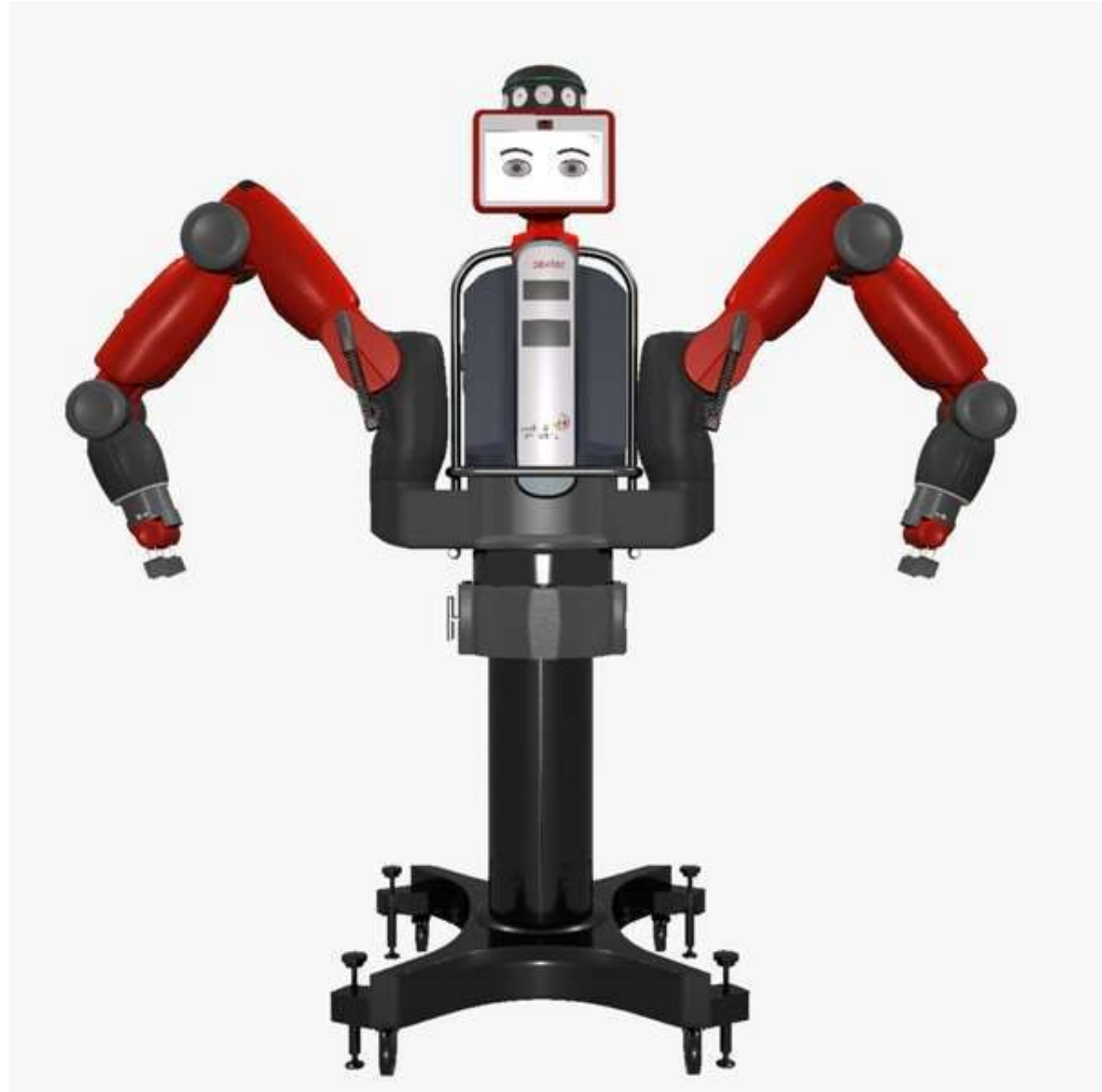


Rethink
Robotics

algus 2008

käivitaja
Rodney Brooks

lõpetas 2018



Rodney Brooks:

„Despite some impressive lab demonstrations we have not actually seen any improvement in widely deployed robotic hands or end effectors in the last 40 years“

A robot that can carry out the last 10 yards of delivery, getting from a vehicle into a house and putting the package inside the front door?

Prediction:

- Lab demo: 2025
- Deployed systems: 2028

